What is claimed is:

1. A system for allocating storage of incoming data packets into a memory of a packet processor, comprising:

a first facility mapping a first block of memory of a fixed block size in bytes into an ordered plurality of atomic pages comprising each a fixed byte size;

a second facility mapping the same block of memory into ordered virtual pages of different sizes, ranging from a smaller virtual page size equal to the atomic page size up to a larger virtual page size equal to the fixed block size;

a third facility to allocate virtual pages as unavailable for storage or de-allocate virtual pages as available for storage;

a fourth facility to receive a data packet, ascertain packet size for the received packet, and to determine fit by checking allocation state for virtual pages of a smallest size that is equal to or larger than the packet size, then allocation state for next larger virtual pages, and so on, until a de-allocated, available virtual page is found; and

a fifth facility to select a virtual page to store the packet, and to update and mark associated atomic pages in the selected virtual page as available or unavailable for storage, in an ordered manner;

characterized in that, after each selection by the fifth facility, state of all atomic and virtual pages is updated.

2. The system of claim 1 implemented in hardware.

3. The system of claim 2 wherein the second facility maps virtual pages in ascending orders of two from the atomic page size up to the block size.

4. The system of claim 3 wherein the block size is 64 KiloBytes (KB), mapped into 256 atomic pages of 256 Bytes each, and further mapped into 256 virtual pages of 256 bytes each, 128 virtual pages of 512 bytes each, and ascending in powers of two to two virtual pages of 32 KB each, and one virtual page of 64 KB.

5. The system of claim 2 further comprising a mechanism for enabling groups of virtual pages by size, wherein the fifth facility selects only among enabled groups of virtual pages.

6. The system of claim 5 wherein the fifth facility selects a de-allocated virtual page for storing the packet only from the enabled virtual page group of the smallest size that is still equal to or larger than the packet size.

7. The system of claim 5 further comprising a second block of memory of the same fixed block size as the first block of memory, mapped in the same way as the first block of memory, wherein a block is selected for storage based on state of enabled virtual page groups in each block, and then a virtual page is selected in the selected block based on fit.

8. The system of claim 7 wherein enabled groups of virtual pages are mapped as enabled in an ascending order from a first block to a last block, having the effect of reserving lower-order blocks for smaller packet size.

9. A data packet router comprising:

external ports to receive and send data packets from and to neighboring connected routers; and

a packet processor having an on-board memory, and comprising a system for allocating storage of data packets in the on-board memory, the system having a first facility mapping a first block of memory of a fixed block size in bytes into an ordered plurality of atomic pages comprising each a fixed byte size, a second facility mapping the same block of memory into ordered virtual pages of different sizes, ranging from a smaller virtual page size equal to the atomic page size up to a larger virtual page size equal to the fixed block size, a third facility to allocate virtual pages as unavailable for storage or de-allocate virtual pages as available for storage, a fourth facility to receive a data packet, ascertain packet size for the received packet, and to determine fit by checking allocation state for virtual pages of a smallest size that is equal to or larger than the packet size, then allocation state for next larger virtual pages, and so on, until a de-allocated, available virtual page is found, and a fifth facility to select a virtual page to store the packet, and to update and mark associated atomic pages in the selected virtual page in an ordered manner.

10. The router of claim 9 wherein the system for storing is implemented in hardware.

11. The router of claim 10 characterized in that, after each selection by the fifth facility, state of all atomic and virtual pages is updated.

12. The router of claim 11 wherein the second facility maps virtual pages in ascending orders of two from the atomic page size up to the block size.

13. The router of claim 12 wherein the block size is 64 KiloBytes (KB), mapped into 256 atomic pages of 256 Bytes each, and further mapped into 256 virtual pages of 256 bytes each, 128 virtual pages of 512 bytes each,

and ascending in powers of two to two virtual pages of 32 KB each, and one virtual page of 64 KB.

14. The router of claim 11 wherein the hardware system further comprises a mechanism for enabling groups of virtual pages by size, and wherein the fifth facility selects only among enabled groups of virtual pages.

15. The router of claim 14 wherein, in the hardware system the fifth facility selects a de-allocated virtual page for storing the packet only from the enabled virtual page group of the smallest size that is still equal to or larger than the packet size.

16. The router of claim 14 wherein the on-board memory further comprises a second block of memory of the same fixed block size as the first block of memory, mapped in the same way as the first block of memory, wherein the hardware system selects a block for storage based on state of enabled virtual page groups in each block, and then a virtual page is selected in the selected block based on fit.

17. A method for allocating storage of data packets in a memory of a packet processor, comprising the steps of:

(a) mapping, by a first facility, a first block of memory of a fixed block size in bytes into an ordered plurality of atomic pages comprising each a fixed byte size;

(b) mapping, by a second facility, the same block of memory into ordered virtual pages of different sizes, ranging from a smaller virtual page size equal to the atomic page size up to a larger virtual page size equal to the fixed block size;

(c) allocating, by a third facility, virtual pages as unavailable for storage or de-allocating virtual pages as available for storage;

(d) receiving a data packet by a fourth facility, ascertaining packet size for the received packet, and determining fit by checking allocation state for virtual pages of a smallest size that is equal to or larger than the packet size, then allocation state for next larger virtual pages, and so on, until a de-allocated, available virtual page is found; and

(e) selecting a virtual page by a fifth facility, to store the packet, and updating and marking associated atomic pages in the selected virtual page in an ordered manner.

18. The method of claim 17 wherein, in step (b), the second facility maps virtual pages in ascending orders of two from the atomic page size up to the block size.

19. The method of claim 17 wherein the block size is 64 KiloBytes (KB), mapped into 256 atomic pages of 256 Bytes each, and further mapped into 256 virtual pages of 256 bytes each, 128 virtual pages of 512 bytes each, and ascending in powers of two to two virtual pages of 32 KB each, and one virtual page of 64 KB.

20. The method of claim 17 further comprising a mechanism for enabling groups of virtual pages by size, wherein the fifth facility selects only among enabled groups of virtual pages.

21. The system of claim 20 wherein the fifth facility selects a de-allocated virtual page for storing the packet only from the enabled virtual page group of the smallest size that is still equal to or larger than the packet size.

22. The system of claim 20 further comprising a second block of memory of the same fixed block size as the first block of memory, mapped in the same way as the first block of memory, wherein a block is selected for storage based on state of enabled virtual page groups in each block, and then a virtual page is selected in the selected block based on fit.

23. The system of claim 22 wherein enabled groups of virtual pages are mapped as enabled in an ascending order from a first block to a last block, having the effect of reserving lower-order blocks for smaller packet size.